This is a peer-reviewed, accepted author manuscript of the following conference paper:
Afsar, M. Z., Stirrat, S. A., & Kokkinakis, I. W. (2020). Investigation of fast GPU-based
algorithms for jet-surface interaction noise calculations.. Paper presented at
2020 AIAA Aviation and Aeronautics Forum and Exposition. https://doi.org/10.2514/6.2020-2564

Investigation of fast GPU-based algorithms for jet-surface interaction noise calculations

M. Z. Afsar *, S. A. Stirrat† and I. W. Kokkinakis‡
Department of Mechanical & Aerospace Engineering, University of Strathclyde, Glasgow, G1 I XJ, UK.

The canonical problem of a jet flow interacting with a plate positioned parallel to the level curves of the streamwise mean flow has received much attention in Aero-acoustics research community as a representation of jet installation effects. Rapid-distortion theory (RDT) modeling of this scenario involves relating an upstream convected quantity that serves as the problem input to measureable turbulence and then determine the far field radiated sound, as the response to this. The latter is found by solving the resulting Wiener-Hopf problem on a discontinuous surface subject to a gust-induced boundary condition across the vortex sheet shed of the trailing edge. Goldstein, Leib & Afsar (J. Fluid Mech., Vol. 881, pp. 551-584, 2019) find that the acoustic spectrum for the round jet scattering problem is given a formula that involves the computation of 4 integrals. Two of these are required to be computed at each point of the two-dimensional domain at a given frequency. Additionally, nested within these integrals is a Fourier transform of the turbulence correlation R_{22}. In GLA19 this Fourier transform was found analytically however for different approximations of R_{22} this isn't possible and it needs to be found numerically. Computation of this form of the solution is naturally computationally expensive on standard desktop computers. In this paper we therefore devise and investigate various algorithms in which the integrals are solved numerically on a GPU card. In general our calculations using the GPU algorithm show considerable reduction in computational time thus making this approach a viable option for design/optimization calculations aimed at characterizing the acoustic signature.

I. Introduction

The recent development in jet-surface interaction noise modeling by Goldstein, Leib & Afsar [1] (hereafter referred to as GLA19) has considered the application of non-homogeneous Rapid-distortion theory to model the low frequency sound amplification produced when a high speed axisymmetric round jet interacts with the trailing edge of a flat plate. The plate in this model is positioned parallel to the level curves of the streamwise mean flow in order to (among other things) mimic the experiment by Bridges[2]. But this arrangement has also received much attention in Aero-acoustics research community because of its canonical representation for the noise radiated by an installed jet.

Rapid-distortion theory (RDT) itself uses linearized equations to analyze rapid changes in turbulent flows such as those that occur when the flow interacts with solid surfaces. It applies whenever the turbulence intensity is small and the length (or time) scale over which the changes take place is short compared to the length (or time) scale over which the turbulent eddies evolve. Asymptotically, these assumptions imply, among other things, that it is possible to identify a distance that is very (infinitely) large on the scale of the interaction, but still small on the scale over which the turbulent eddies evolve. The assumptions also imply that the resulting flow is inviscid and non-heat conducting and is, therefore, governed by the Linearized Euler Equations, i.e., the Euler equations linearized about an arbitrary, usually steady, solution (the base flow) to the nonlinear equations. An important consequence of the disparate length scales is that upstream boundary conditions (specified by two convected quantities that are arbitrary functions of their arguments) can be imposed infinitely far upstream in a region where the flow is undisturbed by the interaction.

The two arbitrary convected quantities do not decay at upstream infinity and can, therefore, be determined from these conditions. Since they are arbitrary functions of their arguments, they can be used as the upstream input in a boundary value problem that seeks to determine the acoustic spectrum as the downstream response But a major problem still remains that these quantities do not correspond to physically measurable variables and the causal RDT solutions for these variables decay at large upstream distances. However, it was shown in GLA19 that since appropriate gradients of

*Chancellor's Fellow. AIAA Member.
†PhD Candidate.
‡Lecturer.

the convected quantities do not decay at upstream infinity, they can be related to measurable flow variables thereby developing a set of physically realizable upstream boundary conditions for planar mean flows and also flows of arbitrary cross-section. For the arrangement discussed above where the flat plate is parallel to the level curves of the jet mean flow, basic application of non-homogeneous RDT [3, 4] shows that the turbulence enters in the pressure and mass flux fluctuation integral solutions for the above arrangement through only one of the convected quantities thus simpifying the subsequent algebraic development.

Our aim in this paper is to develop an algorithm for the fast numerical computation of the acoustic spectrum formula for the round jet/trailing edge interaction problem worked out in the GLA19 analysis (Eqs. 6.26 & 6.27), as well as further developing the formula to achieve better SPL predictions and a better representation of R_{22}. The GLA19 formulation involves 4 multiple integrals, 2 of which determine the spectrum (i.e. the source function) of the convected quantity $\tilde{\omega}_c(\tau - y_1/U(\boldsymbol{y}_T), \boldsymbol{y}_T)$ that enters the analysis (introduced in §.2) are required to be determined at each point in a 2D domain in a conformally-mapped variable and its dummy, and we show that to get a good representation of R_{22} we must also include a numerical Fourier transform. Evaluation of these multiple set of integrals at any given frequency is obviously time consuming on a desktop machine. The algorithm that we use to speed up the evaluation of the acoustic spectrum involves using a GPU card to calculate the 4/5 integrals. Our preliminary calculations show that that the algorithm works very well in reducing the total computation time at a given frequency. We begin the paper by presenting a short review of basic formalism of the problem based leading to a summary of the GLA19 formula.

II. Summary of Rapid-distortion theory (RDT) formulation

GLA19 show that the pressure fluctuation $p'(\boldsymbol{x},t) = (p - \bar{p})(\boldsymbol{x},t)$ produced at the observation point, $(\boldsymbol{x},t) = \{x_1, \boldsymbol{x}_T, t\} = \{x_1, x_2, x_3, t\}$, by the interaction of the arbitrary convected disturbance $\tilde{\omega}_c(\tau - y_1/U(\boldsymbol{y}_T), \boldsymbol{y}_T)$ with solid surfaces $S(\boldsymbol{y}) = S(y_1, \boldsymbol{y}_T) = S(y_1, y_2, y_3)$ embedded in the transversely sheared mean flow $U(\boldsymbol{y}_T)$of an inviscid, non-heat conducting ideal gas is given by

$$p'(\boldsymbol{x},t) = \int_{-T}^{T} \int_{V_\infty(\boldsymbol{y})} G(\boldsymbol{y},\tau|\boldsymbol{x},t)\tilde{\omega}_c(\tau - y_1/U(\boldsymbol{y}_T), \boldsymbol{y}_T)\, d\boldsymbol{y}\, d\tau, \tag{1}$$

where $\boldsymbol{y} = \{y_1, \boldsymbol{y}_T\} = \{y_1, y_2, y_3\}$ is a Cartesian coordinate system with streamwise and transverse components, y_1 and $\boldsymbol{y}_T = \{y_2, y_3\}$ respectively, $\tilde{\omega}_c(\tau - y_1/U(\boldsymbol{y}_T), \boldsymbol{y}_T)$ can be specified as an upstream boundary condition and $G(\boldsymbol{y},\tau|\boldsymbol{x},t)$ denotes the Green's function that satisfies the inhomogeneous Rayleigh equation:

$$LG(\boldsymbol{y},\tau|\boldsymbol{x},t) = \frac{D_0^3}{Dt^3}\delta(\boldsymbol{y} - \boldsymbol{x})\delta(\tau - t), \tag{2}$$

where $D_0/Dt = \partial/\partial t + U(\boldsymbol{x}_T)\partial/\partial x_1$ and:

$$L(\boldsymbol{y},\tau) \equiv \frac{D_0}{D\tau}\left(\frac{\partial}{\partial y_i}c^2\frac{\partial}{\partial y_i} - \frac{D_0^2}{D\tau^2}\right) - 2\frac{\partial U}{\partial y_i}\frac{\partial}{\partial y_1}c^2\frac{\partial}{\partial y_i}, \tag{3}$$

is the Rayleigh operator. The convective derivative in (2) is $D_0/D\tau = \partial/\partial\tau + U(\boldsymbol{y}_T)\partial/\partial y_1$, the mean flow gradient in (2) is $\partial U(\boldsymbol{y}_T)/\partial y_i$ and $c^2(\boldsymbol{y}_T)$ denotes the mean sound speed.

The density-weighted transverse velocity perturbation $u_\perp$ is defined via: $|\boldsymbol{\nabla} U(\boldsymbol{y}_T)|u_\perp \equiv (\partial U(\boldsymbol{y}_T)/\partial y_i)\tilde{u}_i(\boldsymbol{y},\tau)$ where the transverse mass flux perturbation is $\boldsymbol{u}_T(\boldsymbol{y},\tau) = \rho\{v_2', v_3'\}(\boldsymbol{y},\tau)$ with $\rho = \rho(\boldsymbol{y}_T)$ being the mean flow density and $\{v_2', v_3'\}(\boldsymbol{y},\tau)$ being the actual transverse velocity perturbation. The pseudo-density-weighted transverse velocity perturbation $\tilde{u}_i(\boldsymbol{y},\tau)$ is given by integral solution:

$$\tilde{u}_i(\boldsymbol{x},t) = -\int_{-T}^{T} \int_{V_\infty(\boldsymbol{y})} G_i(\boldsymbol{y},\tau|\boldsymbol{x},t)\tilde{\omega}_c(\tau - y_1/U(\boldsymbol{y}_T), \boldsymbol{y}_T)\, d\boldsymbol{y}\, d\tau, \tag{4}$$

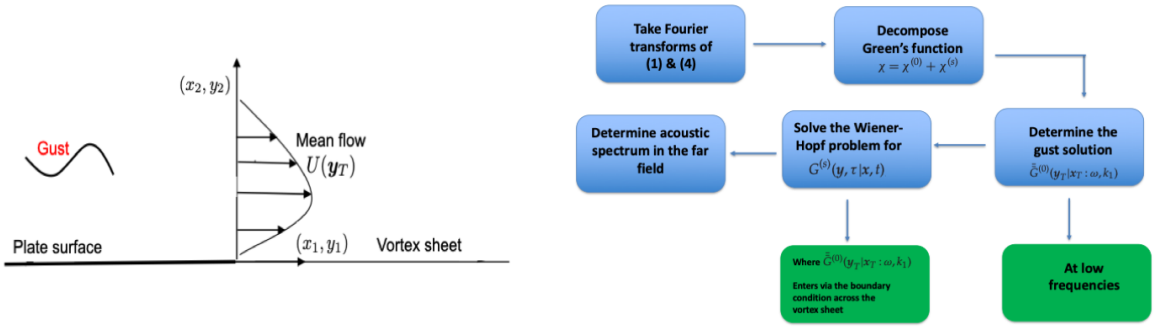
with $G_i(\boldsymbol{y},\tau|\boldsymbol{x},t)$ can be expressed in terms of the three-dimensional gradient of $G(\boldsymbol{y},\tau|\boldsymbol{x},t)$ via

$$\frac{D_0^2 G_i}{D\tau^2}(\boldsymbol{y},\tau|\boldsymbol{x},t) = -\frac{\partial}{\partial y_i}G(\boldsymbol{y},\tau|\boldsymbol{x},t) \tag{5}$$

where $i = (2, 3)$ in (4) and (5).

A. Canonical jet-surface interaction problem

The solid surfaces $S(\boldsymbol{y})$ bound volume $V(\boldsymbol{y})$ in formulae (1) & (4) can be finite, semi-infinite or infinite in the streamwise direction y_1 but its generators must be parallel to the level curves of the mean velocity field. The Green's function, $G(\boldsymbol{y},\tau|\boldsymbol{x},t)$, now satisfies the homogeneous boundary condition $\Gamma(\boldsymbol{y},\tau|\boldsymbol{x},t) = 0$ for $y \in S$ where the scalar field, Γ, is determined to within an arbitrary convected quantity by the boundary condition, $D_0^2\Gamma(\boldsymbol{y},\tau|\boldsymbol{x},t)/D\tau^2 \equiv n_j c^2 \partial/\partial y_i G(\boldsymbol{y},\tau|\boldsymbol{x},t)$ which reduces to the usual zero normal derivative boundary condition on the (impermeable) plate surface $S(\boldsymbol{y})$ present in the flow for the canonical RDT problem in Fig. 1. $\boldsymbol{n}(\boldsymbol{y}_T)$ is the outward pointing unit normal, $\boldsymbol{n}(\boldsymbol{y}_T) = \boldsymbol{\nabla} U(\boldsymbol{y}_T)/|\boldsymbol{\nabla} U(\boldsymbol{y}_T)|$. Downstream of the trailing edge in Fig. 1, the Green's function $G(\boldsymbol{y},\tau|\boldsymbol{x},t)$ must satisfy the jump conditions $\triangle[G] = \triangle[\Gamma] = 0$ for $\boldsymbol{y}_T \in S_0$ across the resulting downstream wakes (or vortex sheets) where S_0 denotes the surfaces of discontinuity and $\triangle[\bullet]$ denotes the jump in $\bullet$ across these surfaces. The mean velocity profiles can be discontinuous across the wakes which can then support additional spatially growing instability waves that can be generated by imposing a Kutta condition at the trailing edge or suppressed by imposing a boundedness requirement.



(a) Canonical Rapid-distortion theory problem

(b) Solution process

Fig. 1 Rapid-distortion theory (RDT) modeling for the determination of trailing edge noise.

The simplest way to determine the acoustic radiation using (1) is to split the Green's function up into a hydrodynamic component (that does not generate any acoustic waves at subsonic Mach numbers) and a non-hydrodynamic component that corresponds to the acoustic waves that propagate to the far-field when inserted into (1). Mathematically, this can be accomplished by dividing the Rayleigh equation Green's function that appears in the solution (1) into two components: $G(\boldsymbol{y},\tau|\boldsymbol{x},t) = G^{(0)}(\boldsymbol{y},\tau|\boldsymbol{x},t) + G^{(s)}(\boldsymbol{y},\tau|\boldsymbol{x},t)$ where $G^{(0)}(\boldsymbol{y},\tau|\boldsymbol{x},t)$ denotes a particular solution of (2) which can either be defined on all space or, be required to satisfy appropriate boundary conditions on a streamwise extension (i.e. the vortex sheet) of the bounding surface $S(\boldsymbol{y})$ that extends from minus to plus infinity in the streamwise direction. This decomposition implies $G_i(\boldsymbol{y},\tau|\boldsymbol{x},t)$ in (4) also decomposes in a similar way and therefore the pressure fluctuation in (1), decomposes as $p'(\boldsymbol{x},t) = p'^{(0)}(\boldsymbol{x},t) + p'^{(s)}(\boldsymbol{x},t)$ where $p'^{(0)}(\boldsymbol{x},t)$ which is given by (1) and (2) with $G(\boldsymbol{y},\tau|\boldsymbol{x},t)$ replaced by $G^{(0)}(\boldsymbol{y},\tau|\boldsymbol{x},t)$, does not produce any acoustic radiation at subsonic Mach numbers and can, therefore, be identified with the hydrodynamic component of the unsteady motion. On the other hand, the 'scattered component', $G^{(s)}(\boldsymbol{y},\tau|\boldsymbol{x},t)$, satisfies the homogeneous Rayleigh's equation along with appropriate inhomogeneous boundary and jump conditions on the streamwise discontinuous surfaces $S(\boldsymbol{y})$ and $S_0(\boldsymbol{y})$. The corresponding 'scattered solution', $p'^{(s)}(\boldsymbol{x},t)$ therefore, accounts for all of the acoustic components of the motion.

B. GLA19 acoustic spectrum formula

GLA19 used the theory described above to determine the acoustic spectrum due to $p'^{(s)}(\boldsymbol{x},t)$. Using appropriate asymptotic analysis to determine both $G^{(0)}(\boldsymbol{y},\tau|\boldsymbol{x},t)$ and $G^{(s)}(\boldsymbol{y},\tau|\boldsymbol{x},t)$ (see App. C of their paper) they obtained the

following formula valid at low frequencies:

$$I(\boldsymbol{x},\omega) \sim \int_{-\infty}^{0}\int_{-\infty}^{0} D(u,\tilde{u};\theta)\bar{S}(u,\tilde{u};k_3^*,\omega)\,du\,d\tilde{u}, \tag{6}$$

(Eqs. 6.26 & 6.27 in their paper) for an arbitrary mean flow has level curves that are relatively concentric inasmuch as $U(y_2,y_3) = U(u(y_2,y_3))$ where u is the real part of the conformal mapping given by (5.1) in GLA19. $D(u,\tilde{u};\theta)$ is then the round jet directivity factor determined by application of the Wiener-Hopf technique (i.e. Eq. 13 in Afsar *et al* (2017).[5] written in $(u,\tilde{u})$ co-ordinates where $\tilde{u}$ is another location along the level curve $u = const$). While $D(u,\tilde{u};\theta)$ is an algebraic function of $(u,\tilde{u})$, the integrated spectrum of $\tilde{\omega}_c$, $\bar{S}(u,\tilde{u};k_3^*,\omega)$ is determined

$$\bar{S}(u,\tilde{u};k_3^*,\omega) = \int_{-\infty}^{0}\int_{-\infty}^{0} S(u,\tilde{u}|v,\tilde{v};k_3^*,\omega)\left|\frac{dz}{dW}\right|^2\left|\frac{d\tilde{z}}{d\tilde{W}}\right|^2 dv\,d\tilde{v}, \tag{7}$$

where $v(y_2,y_3) = const.$ are the orthogonal curves to $u(y_2,y_3) = const.$ defined by the Cauchy-Riemann equations and $d\tilde{z}/d\tilde{W}$ is the appropriate inverse Jacobian function between the mapping $z(y_2,y_3) \rightarrow W(u,v)$) that is made specific in App. A in GLA19 and is also an algebraic function of co-ordinates, (u,v). The spectrum $S(u,\tilde{u}|v,\tilde{v};k_3^*,\omega)$ is modelled and is shown by GLA19 to be proportional to the modified Bessel function of the second kind, $K_1(...)$. Since the numerical computation of (6) requires calculation of (7) at each point in the $(u,\tilde{u})$-domain, the acoustic spectrum calculation is potentially quite computationally expensive and requires a call to a Bessel function library to evaluate $K_1(...)$. In the next section we therefore discuss the application of a Graphics Processing Unit (GPU) based algorithm to speed up the calculation process. Our results, which start out using a generic set of nested integrals, indicate in general that if more information of the turbulence is used to define the spectrum $S(u,\tilde{u}|v,\tilde{v};k_3^*,\omega)$, a GPU-based algorithm becomes advantageous in effectively speeding up the calculation. By considering the upstream asymptote of the Fourier transform of the evolution equation for the convected quantity, $\tilde{\omega}_c$, in the arbitrary curvilinear co-ordinates (u,v) (that are made definite in this particular problem by using a conformal mapping as indicated above), GLA19 showed that the function $S(u,\tilde{u}|v,\tilde{v};k_3^*,\omega)$ is defined by

$$S(u,\tilde{u};v,\tilde{v}) = l_2^4 A(u,\tilde{u})(\rho_\infty c_\infty^2)^2\left[\frac{dU/du}{U^2(u)}\frac{dU/d\tilde{u}}{U^2(\tilde{u})}|\nabla u|\left|\tilde{\nabla}\tilde{u}\right|\omega^2\right] \times \frac{\tau_0 f}{\pi\sqrt{1+\tilde{\omega}^2}}K_1(f\sqrt{1+\tilde{\omega}^2}). \tag{8}$$

This result shows, among other things, that $S(u,\tilde{u}|v,\tilde{v};k_3^*,\omega)$ is directly proportional to the Fourier transform of the streamwise-independent transverse velocity correlation function, R_{22}, for which considerable turbulence data exists (see Afsar *et al*. 2017)[5]. where the amplitude function $A(u,\tilde{u})$ was taken as

$$A(u,\tilde{u}) = A_0\sqrt{(u\tilde{u})^5(dU/du)(dU/d\tilde{u})\left|dW/dz\right|_{v=0}\left|d\tilde{W}/d\tilde{z}\right|_{\tilde{v}=0}} \tag{9}$$

and the jet mean flow near the plate by the function: $U(u) = U_d(1-e^{-\kappa u^2-\kappa_1 u^4})$. While (A_0,κ,κ_1) are parameters to be tuned they are kept fixed for a particular jet/plate configuration and acoustic Mach number. Ideally these parameters are found by comparison to experimental data as was done in the rectangular jet problem by Goldstein *et al*. (GLA17, p.499)[4]. We kept these values identically the same as that used in GLA19 (see p. 572 & Fig. 5). The remaining free parameters in Eq. (8) are then related to the turbulence structure of R_{22}, i.e., (l_2,l_3,τ_0). Note that $\tilde{\omega}$ is the $O(1)$ scaled frequency, defined by $\tilde{\omega} = \omega\tau_0 = O(1)$ and the length scales (l_2,l_3) enter the transverse spectral decay function, f; we kept this the same as GLA19 (see p. 572) after experimenting with various alterations thereof (Afsar *et al*., 2020)[6].

III. Extension of the GLA19 result and various analytical approximations

For convergence to 2dp we discovered that the GLA19 model was not computationally expensive since not many iterations were needed for each loop. However, in this section we will discuss an extension to the GLA19 model which increases the complexity of the program and takes longer to run but which greatly improves the SPL predictions.

A. Anti-correlation effects

We found that including an extra term in the structure of the R_{22} correlation function that enters (8) via the algebraic relation (4.21) in GLA19 and which is needed to capture the anti-(negative) correlation gave better agreement at all

frequencies of interest for the range of flow speeds we considered. For example, at $Ma = 0.5$, the agreement is better at high frequency compared to GLA19. At St 1, the agreement is almost $(5-8)dB$ closer to the acoustic data than GLA19. On the other hand, at $Ma = 0.9$, the agreement is closer to within $10dB$ compared to the GLA19 predictions. Both of these cases are at the observation angle of $\theta = 90°$ compared to the jet centre-line. In Afsar *et al.* (2020)[6], we show that the term $\tau_0 f K_1(f\sqrt{1+\tilde{\omega}^2})/(\pi\sqrt{1+\tilde{\omega}^2})$ in (8) is replaced with

$$(1-a_1)\frac{\tau_0 f}{\pi\sqrt{1+\tilde{\omega}^2}}K_1(f\sqrt{1+\tilde{\omega}^2}) + \frac{a_1\tau_0 f^2\tilde{\omega}^2}{\pi(1+\tilde{\omega}^2)}\Big[\frac{1}{2}(K_0(f\sqrt{1+\tilde{\omega}^2}) + K_2(f\sqrt{1+\tilde{\omega}^2})) + \frac{K_1(f\sqrt{1+\tilde{\omega}^2})}{f\sqrt{1+\tilde{\omega}^2}}\Big] \tag{10}$$

when a term of the form of $1 - a_1\hat{\tau}\partial/\partial\hat{\tau}$ acts upon the exponentially decaying de-correlation function in (B.5) of GLA19, where $\hat{\tau} = \tilde{\tau} - [y_1/U(u) - \tilde{y}_1/U(\tilde{u})]$ in (B.5) and Fourier transform (4.21) of their paper. We determined the parameters that enter this model in two ways. First by hand tuning to get good agreement against the acoustic data (Brown 2013)[7] and secondly by using an optimization routine (see Afsar *et al.* 2019)[8] to determine the most appropriate values for these parameters by comparison to turbulence data for the auto-correlation of R_{22} (Fig. 12 in Bridges, 2006)[9] for SP07 (Cold jet at $Ma = 0.9$). Assuming that the the temporal decay and the anti-correlation parameters are relatively constant with reduction in Ma, we found that the optimised parameters were: $a_1 = 0.889, \tau_0 = 5.0$. Since the turbulence data in Bridges' paper did not consider the transverse spatial separations in the measurement of R_{22}, we could only optimize against the auto-correlation of the latter function. This meant estimating the length scale parameters (l_2, l_3). Our numerical experiments showed that allowing the spanwise length scale parameter to remain fixed across the Ma range (i.e. at the value of $l_3 = 5.0$) produced the most reasonable predicted acoustic spectrum. The only 'free' parameter in the problem is then the turbulence length scale in the wall normal direction. We found that this had to reduce with reduction in Ma indicating that the reduction in absolute magnitude of edge-related noise. Hence we found that $l_2 = 0.5$ for $Ma = 0.5$, $l_2 = 0.9$ for $Ma = 0.7$, and $l_2 = 1.2$ for $Ma = 0.9$. These parameters were also tested at observation angles other than the standard case of $\theta = 90°$ and were found to be give results that were in reasonable agreement and importantly closer to the acoustic data than the GLA19 predictions where the turbulence model was not compared against experimental data (see Afsar *et al.* 2020)[6]. The results for the $\theta = 90°$ are shown in Fig. 2 below.

1. OpenACC *comparison for CPU and GPU calculation*

As mentioned above, for model 1, the problem reduces to a four-dimensional integral with an integrand that includes several Bessel functions. To evaluate these Bessel function the Amos library is used, however, this library is outdated, written in Fortran 77 and takes a long time to run. Profiling led to the discovery that roughly 97% of computation time was spent either within zbesk or the subroutines that it calls from the Amos library. Therefore we decided to parallelize the code for the integral assuming (for now) that the argument of $K_1(...)$ depends on v or $\tilde{v}$.

The program was written using OpenACC directives to enable compilation for either a multicore CPU or offload to a GPU. We compare the speed increase between GPU and multicore for two programs: one integral and two integrals. Ultimately the same process will be used for the four integral program but for now we test how fast a single integral code will take to run to evaluate an integrand with a Bessel function of type, $K_1(...)$, that is called via Amos. The general program structure for the three programs are shown in Fig. 3. All programs use Simpson's rule to carry out the integration of an analytical test case. This is to ensure results are correct and that OpenACC can be optimised before applying it to the much more complicated real function. To fix the test case, we consider the Fourier transform of a first order modified Bessel function which, as shown in Table 1, can be integrated analytically[10]. Fig. 4 and 5 show the program architecture for multi-core (6 cores) and offload to GPU respectively. This highlights the fact that offloading to the GPU increases the number of calculations that can be done in parallel however it requires copying information to and from the GPU memory, which can be time intensive.

The time taken to run the one integral program was initially found to be longer in the GPU than multicore. This was when a NVIDIA GeForce GT 710 card was used, and was likely due to the limitation of computation power for double precision calculation (122.1 GFlops) and the fact that the card was also being used to display graphics on the monitor. As shown in Fig. 6 running the program on GPU cards built for computation exhibit a large speed increase.

The test function for the double integral calculation (mimicking the $(v, \tilde{v})$ integrals in 7) is shown in Table 2, along with the analytical result. Like the one integral test case, Fig. 7 and 8 show the program architecture for multi-core (6 cores) and offload to GPU respectively. It should be highlighted that only the inner loop has been made parallel at this point.

The speed up for the 2-integral case is shown in Fig. 9. Evidently, this speed up scales much better for the GPU than the 1-integral did. This could be due to the first initialization being more costly than the subsequent ones (in the 2d case).

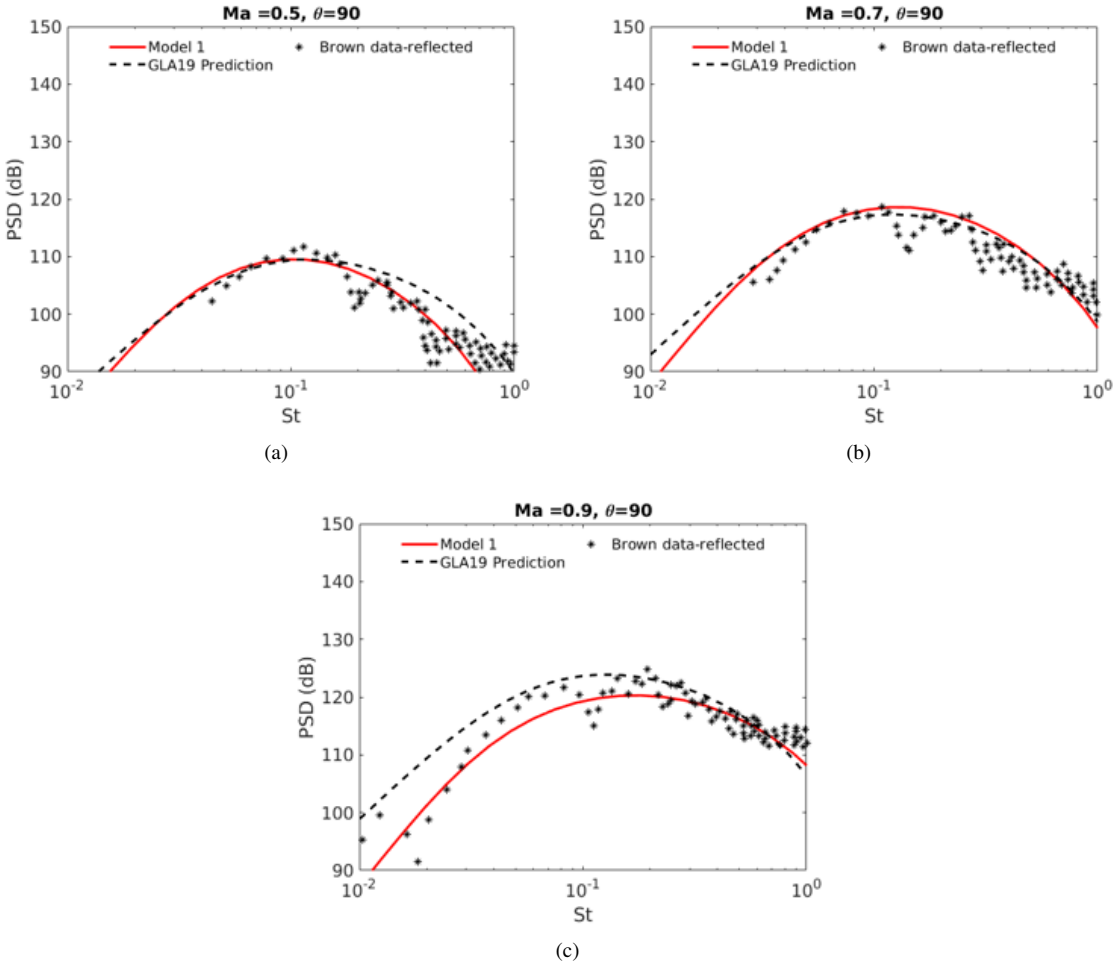


Fig. 2 SPL prediction against acoustic data in [7]. SPL computed via $SPL = 10log_{10}4\pi(\rho U_J^2)^2 I(\boldsymbol{x};\omega)/P_{ref}^2$ where $P_{ref} = 2\times10^{-5}$ Pa. The spectrum $I(\boldsymbol{x};\omega)$ is determined by integrating (6) over $(u, \tilde{u})$ using (10) in (8).

2. Adapting the gpu algorithm for model 1

So far, we have focused on parallelizing the innermost loop, however this means that data needs to be copied into and out of the ACC parallel region for every value of u, which is inefficient. Therefore we will move the ACC region to the outer loop. We were initially going to parallelize both loops through the use of gangs and vectors to maximise GPU usage and further speed up the program. The two integral program architecture for this is displayed in Fig. 10. Likewise we first developed the program architecture for the 4-integral in a similar manner and tested it using the analytical test function shown in Table 3 (as mentioned the latter is advantageous because we can evaluate analytically a series of nested integrals with one involving a Bessel function) and used this as a precursor to doing the full integral in Eqs. (6) & (7). We were initially going to parallelize the four integral program over gangs, workers and vectors as shown in Fig. 11. However, we then realised that the compiler would only parallelize the outer loop over gangs and vectors. All other loops were best run sequentially. The program architecture for this is shown in Fig. 12.

However, we discovered that when using the R_{22} for model 1 not enough iterations were needed to fully utilise the GPU. Fig. 13 shows that so few iterations are needed for convergence to 0dp that offloading to the GPU actually slows down the code since the initialisation of the GPU takes longer than the code takes to run. It also shows that even for 2dp convergence, speed up is low due to number of iterations not fully utilising the GPU and the complexity of the code meaning each iteration does not take long to compute.

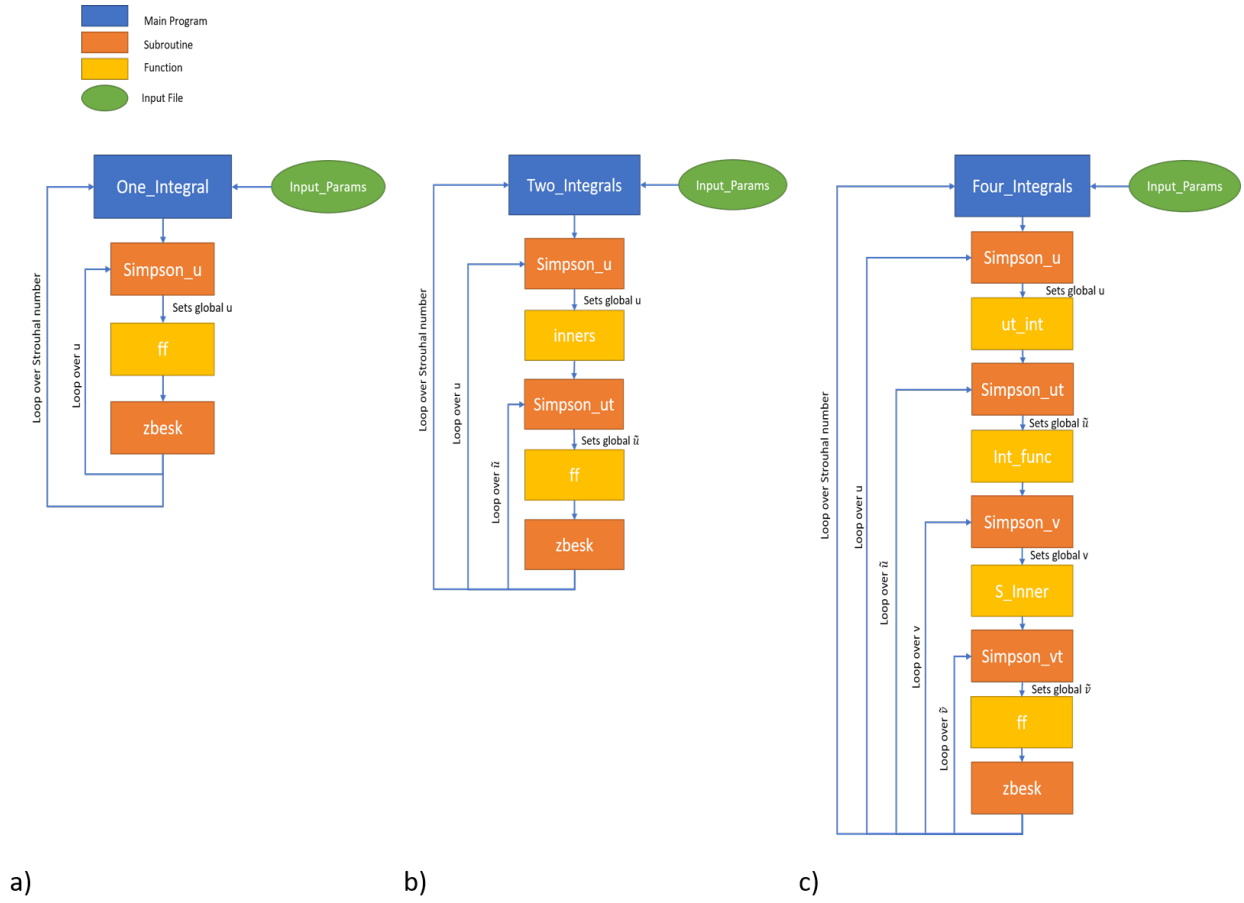


Fig. 3 General program structure for a) one integral, b) two integrals, and c) four integrals

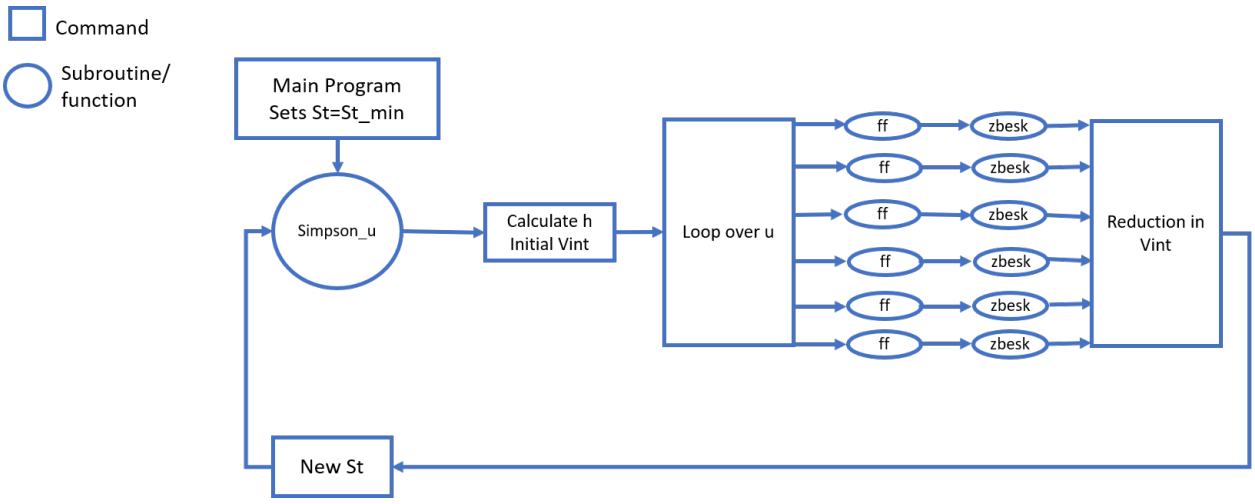


Fig. 4 One Integral: Program architecture for multi-core (6 core)

B. Limiting form of the GLA19 result at infinite spanwise length scale, $l_3 \to \infty$

In Afsar *et al.* (2020)[6] we approximate (4.22) in GLA19 when the spanwise length scale is long, inasmuch as $l_3 \to \infty$ compared to the $O(1)$ wall normal scale, l_2. The $(v, \tilde{v})$ integrals in (7) can be worked out analytically in this

Function to Integrate	$G(\omega) = \int_{-\infty}^{\infty} K_1(2\pi ui + 1)\exp(i2\pi\omega u)du$
Analytical Result	$\frac{\omega \exp(-\omega)}{\sqrt{(\omega^2-1)}}$

Table 1 One Integral: Analytical Test Function

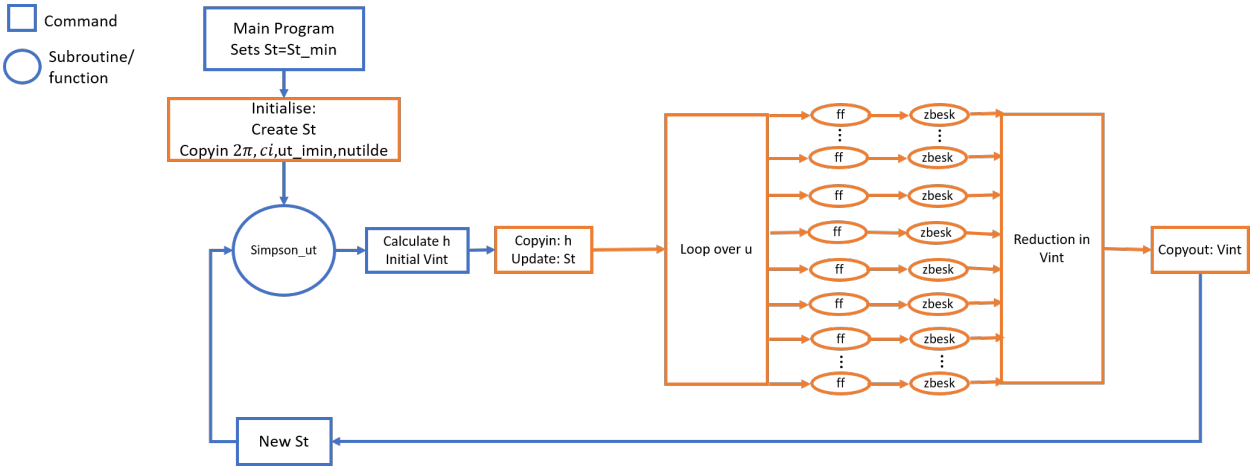


Fig. 5 One Integral: Program architecture for offload to gpu (blue = cpu instruction, orange = gpu instruction)

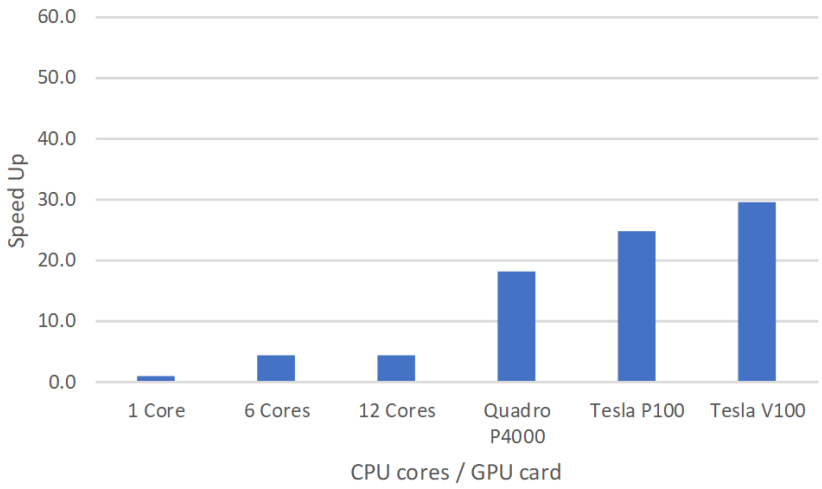


Fig. 6 One Integral: Speed up by using openacc on the loop

limit. We find that (6) reduces to

$$I_\omega = \left(\frac{k_\infty}{4\pi|\bar{x}|}\right)^2 \frac{4\pi^2\tau_0\omega^2}{\pi\sqrt{1+\tilde{\omega}^2}}(\rho_\infty c_\infty^2)^2 \int_{-\infty}^{0}\int_{-\infty}^{0} \frac{D(u,\tilde{u};\theta)Q(u)Q(\tilde{u})}{\sinh(u)\sinh(\tilde{u})} l_2^4 A(u,\tilde{u}) \left[\frac{dU/du}{U^2(u)}\frac{dU/d\tilde{u}}{U^2(\tilde{u})}\right] f(u,\tilde{u}) K_1(f\sqrt{1+\tilde{\omega}^2}) d\tilde{u}du \tag{11}$$

Function to Integrate	$G(\omega) = \int_1^2 \int_{-\infty}^{\infty} K_1(2\pi\tilde{u}i + u)\exp(i2\pi\omega\tilde{u})d\tilde{u}du$
Analytical Result	$\frac{1}{\sqrt{(\omega^2-1)}}(exp(-\omega) - exp(-2\omega))$

Table 2 Two Integrals: Analytical Test Function

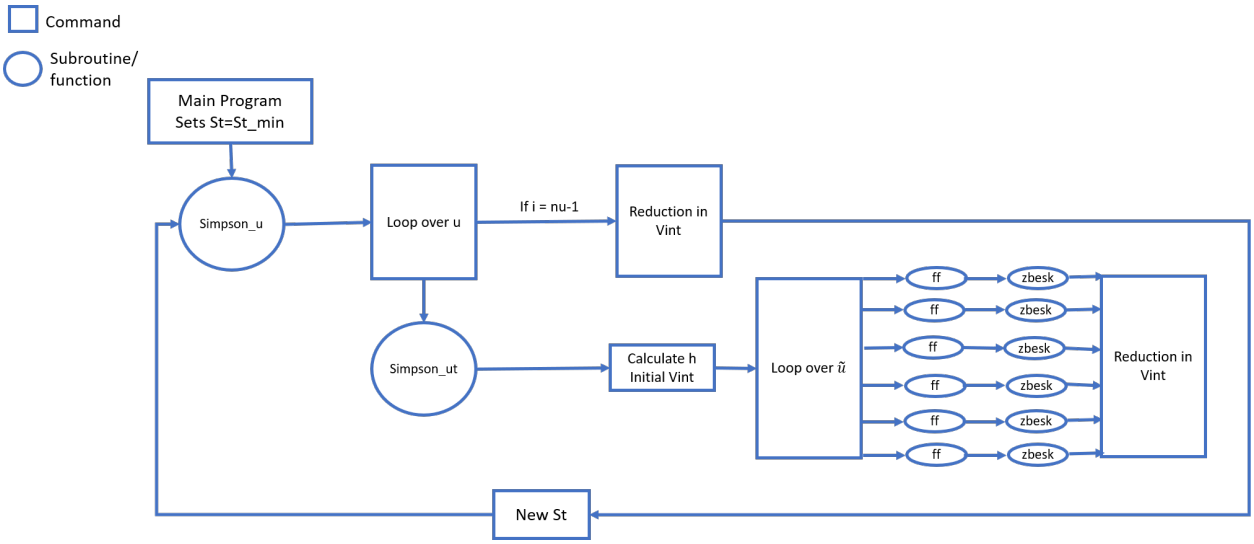


Fig. 7 Two Integrals: Program architecture for multi-core (6 core)

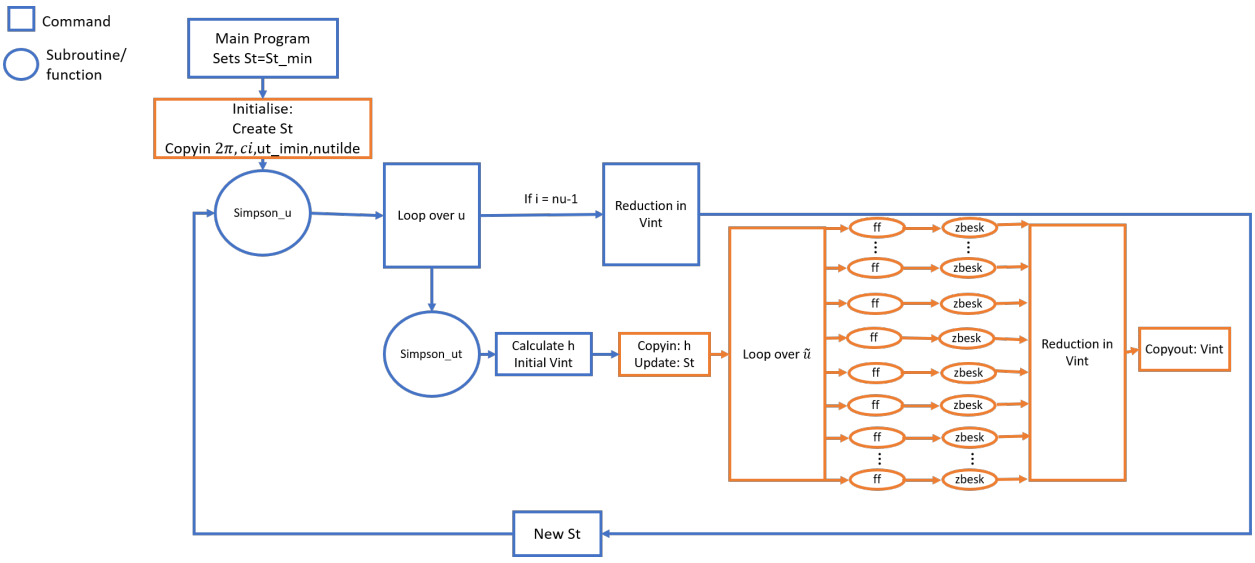


Fig. 8 Two Integrals: Program architecture for offload to gpu (blue = cpu instruction, orange = gpu instruction)

which is now the long-l_3 limit of (6.28) in GLA19. Note that the integrals in (11) must then be interpreted as a Cauchy Principal value owing to the simple pole where $\sinh u \sim u$ as $u \to 0$ (corresponding to the plate surface via A.9 in GLA19) and therefore $\int_{-\infty}^{0}\{\bullet\} \to_{\delta\to 0^+} \int_{-\infty}^{\delta}\{\bullet\}$ in order to obtain a converged integral. However, our calculations

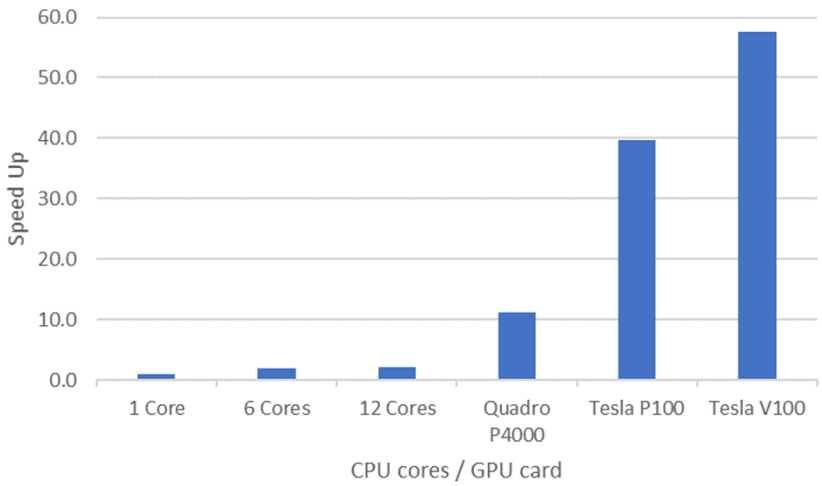


Fig. 9 Two Integrals: Speed up by using openacc on the inner loop

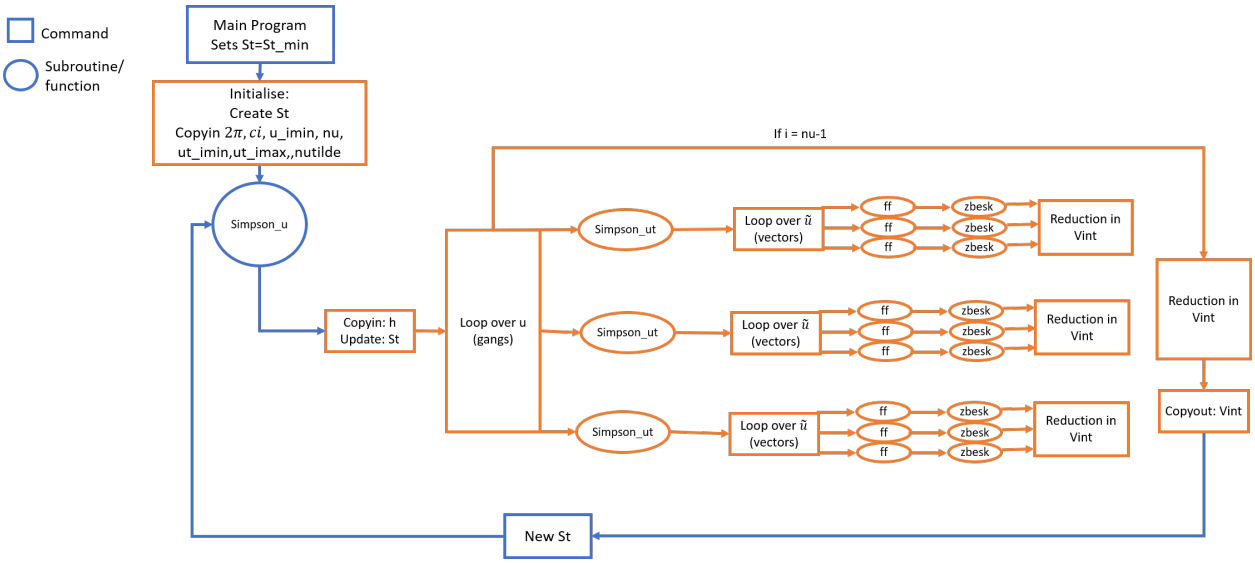


Fig. 10 Two Integrals: Initial program architecture offload to gpu (blue = cpu instruction, orange = gpu instruction) where both loops are parallelized

confirm that integrand in (11) is bounded for all $(u, \tilde{u}) = (-\infty, 0]$ as long as the amplitude function $A(u, \tilde{u})$ is algebraic in $(u, \tilde{u})$ inasmuch as $A(u, \tilde{u}) = u^{m+1}$ when $\tilde{u} = O(1) \neq 0$ where $m \geq 1$ (and vice versa when $u = O(1) \neq 0$ and $\tilde{u} \to 0$). When this approximation is combined with the anti-correlation expansion (10), the resulting predictions are identical to the full numerical evaluation of (6) when (10) is inserted into the integrals over $(\nu, \tilde{\nu})$ in (7) at low frequencies. In other words, *the long-l_3 limit of (6) when anti-correlation effects via (10) are present captures the correct low frequency roll-off of the acoustic spectrum.* Said another way, the low frequency algebraic roll-off of I_ω in (6) scales as ω^n where the value of the exponent is determined by the spanwise independent form of (6). The asymptotic extension of the long l_3 form of (6) to high frequencies follows naturally when use is made of the appropriate WKBJ solution developed in GLA17[4] rather than that used in GLA19 (p.571). Again, further details of this are given in Afsar *et al.* (2020)[6], we simply report the acoustic spectrum predictions here. Fig. 14 shows the predictions made using the analytical formula is more-or-less identical to the full numerical calculation that requires 4-integrals to be determined. Further numerical calculations in Afsar *et al.* (2020)[6] reveal that this conclusion remains true at smaller observation angles to the jet axis where the edge noise makes a reduced contribution to the total installation noise magnitude at any given frequency

Function to Integrate	$G(\omega) = \int_0^1 \int_0^{\pi/2} \int_1^2 \int_{-\infty}^{\infty} K_1(2\pi\tilde{v}i + v) \exp(i2\pi\omega\tilde{v}) cos(\tilde{u}) u d\tilde{v} dv d\tilde{u} du$
Analytical Result	$\frac{1}{2\sqrt{(\omega^2-1)}}(exp(-\omega) - exp(-2\omega))$

Table 3 Four Integrals: Analytical Test Function

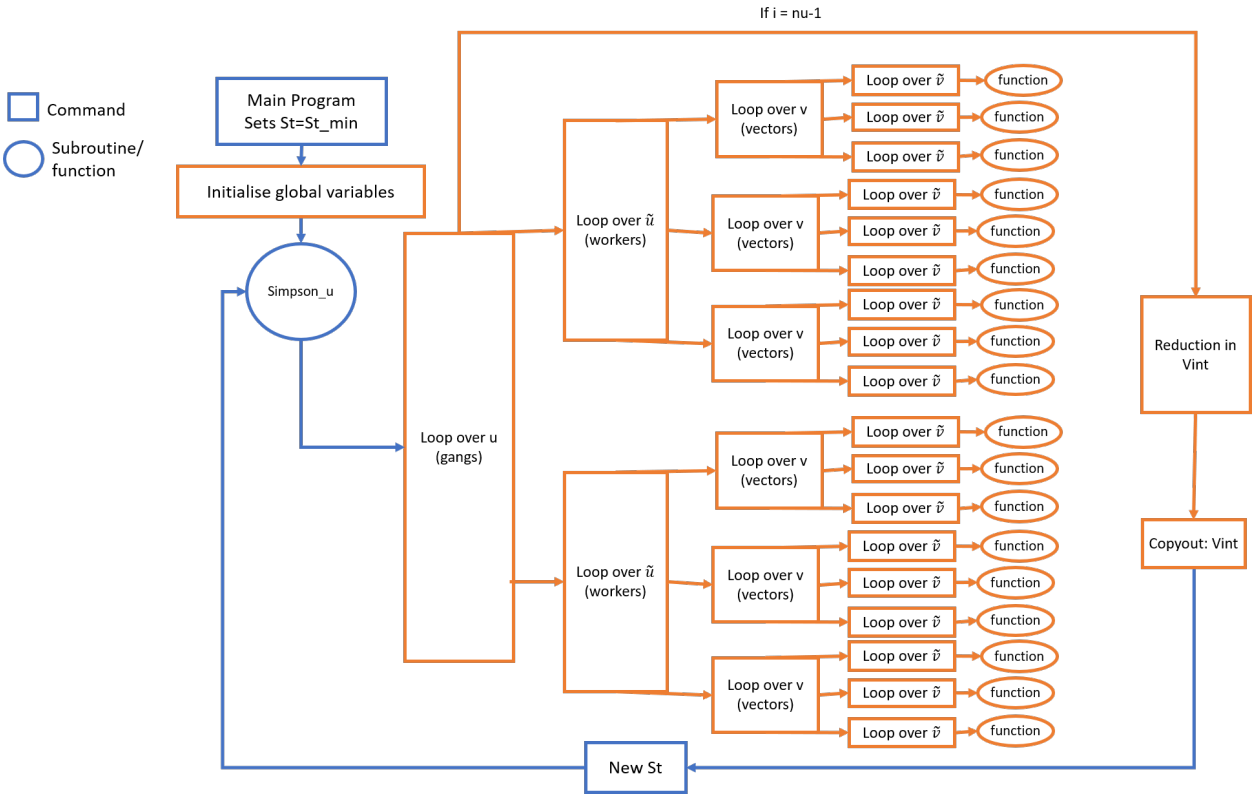


Fig. 11 Four Integrals: Initial program architecture offload to gpu (blue = cpu instruction, orange = gpu instruction), parallelism begins at outer loop

covering the Strouhal number range, $St = [0.01, 1.0]$.

C. Effect of inclusion of finite algebraic oscillations in the R_{22} function

The Bridges' measurements show that the auto-correlation of R_{22} will possess finite-amplitude algebraic oscillations– that is, they appear not to be a small perturbation of the main exponential decay. We attempted to capture this feature of the turbulence structure by developing a set of models of increasing algebraic complexity in terms of the combination of suitably weighted trigonometric/transcendental functions multiplied by an appropriate exponentially decaying function of time delay. Models (A,B,C) in Fig. (15) show

$$\text{Model A} \quad : \quad R_{22} = (1 - a_1 + a_2\hat{\tau}^2 - a_3 \sin(b\hat{\tau}))e^{-\frac{\hat{\tau}^2}{c}} \tag{12}$$

$$\text{Model B} \quad : \quad R_{22} = (1 - a_1 + a_2\hat{\tau}^2 - a_3 \sin(b\hat{\tau}))e^{-\frac{\hat{\tau}^2}{c}} + (a_4 \cos(b_2\hat{\tau}))e^{-\frac{\hat{\tau}^2}{c_2}} \tag{13}$$

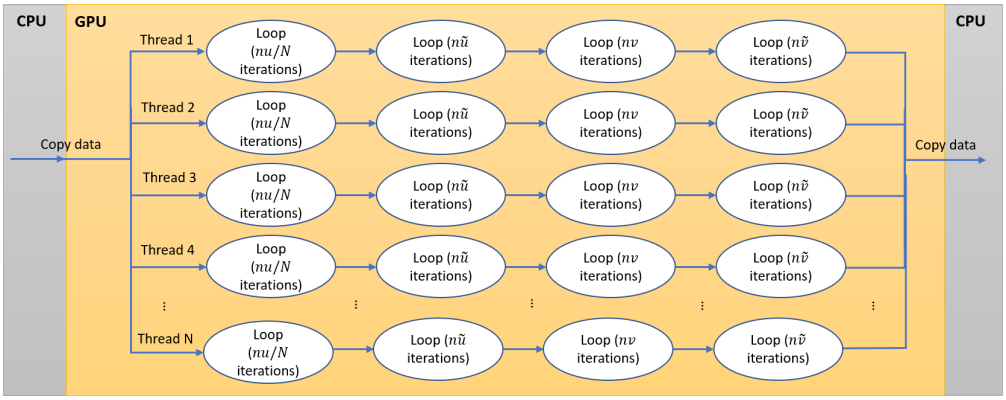


Fig. 12 Four Integrals: Most efficient program architecture, parallelising over gangs and vectors on the outer loop

Model		Spatial Convergence					Time (s)			Speed Up
							Serial		GPU	
		nu	$n\tilde{u}$	nv	$n\tilde{v}$	TOTAL	1 St	30 St	30 St	30 St
Model 1 (a1)	0dp	50	10	10	10	50,000	0.156	4.481	14.342	0.3
	1dp	1000	20	20	10	4,000,000	10.474	313.11	37.808	8.3
	2dp	1000	20	20	20	8,000,000	19.690	593.93	71.971	8.3

Fig. 13 Model 1: speed up for different levels of convergence. $(nu, n\tilde{u}, nv, n\tilde{v})$ are the number of iterations in loops $(u, \tilde{u}, v, \tilde{v})$. Total refers to the total number of iterations across all loops.

$$\text{Model C} \quad : \quad R_{22} = (1 - a_1 + a_2\hat{\tau}^2 - a_3 \sin(b\hat{\tau}))e^{-\frac{\hat{\tau}^2}{c}} + (a_4 \cos(b_2\hat{\tau}))e^{-\frac{\hat{\tau}^2}{c_2}} + (a_5 \cos(b_3\hat{\tau}))e^{-\frac{\hat{\tau}^2}{c_3}} \tag{14}$$

The use of the formulae for models (A,B,C) as a replacement of (B.5) in GLA19 or the anti-correlation dependent function in (10) implies that the Fourier transform given by (4.22) in their paper cannot, in general, be computed analytically. Since the integrand is bounded when functions of the type given in (12–14), (4.22) can be interpreted as a Riemann integral that can be evaluated via Simpson's rule. Our calculation of the latter converged easily however this obviously increases the total number of iterations which can therefore be made to run faster if use is made of the GPU in its numerical evaluation. The model coefficients in (12–14) were found using the optimization routine reported in Afsar *et al.* (2019)[8] and are given in the caption to Fig. (15). Note also that the since only the auto-correlation of R_{22} (i.e. the temporal variation of this function at zero spatial separation in (u, v)-co-ordinate directions), a 'separable' model was used in which the spatial decay was given, as a first step, by a simple exponentially decaying function of the form $e^{-f(u,v)}$.

Although models (A-C) also imply increased code complexity, the potential increase in speed-up is greater even when the number of outer iterations remain at 1000, as shown in Fig. 16 However, the GPU is still not fully utilised, and if we increase the number of outer iterations we get a better speed up, as indicated in Fig. 17. Speed up doubles until it reaches 4000 iterations at which point the GPU is saturated. Beyond this, speed up still increases but at a slower rate until it reaches the maximum of 130. We note that if the number of iterations on the Fourier transform (inner) loop increases, speed up decreases slowly as shown in Fig. 18. If it reaches a point where the number of iterations on the inner loop greatly outweighs those on the outer loop it may be beneficial to only offload the inner loop to the GPU. It should be noted that in doing this the cost of data transfer is much higher since on every outer iterations data will need to be transferred to the GPU and vice versa. Therefore, the number of iterations in the inner loop must vastly outweigh those in the outer loop, as shown in Fig. 19. For this test we set the number of iterations in the u loop to 100, therefore we see it is only when the iterations in the inner loop are 320 times higher that any benefit appears. The

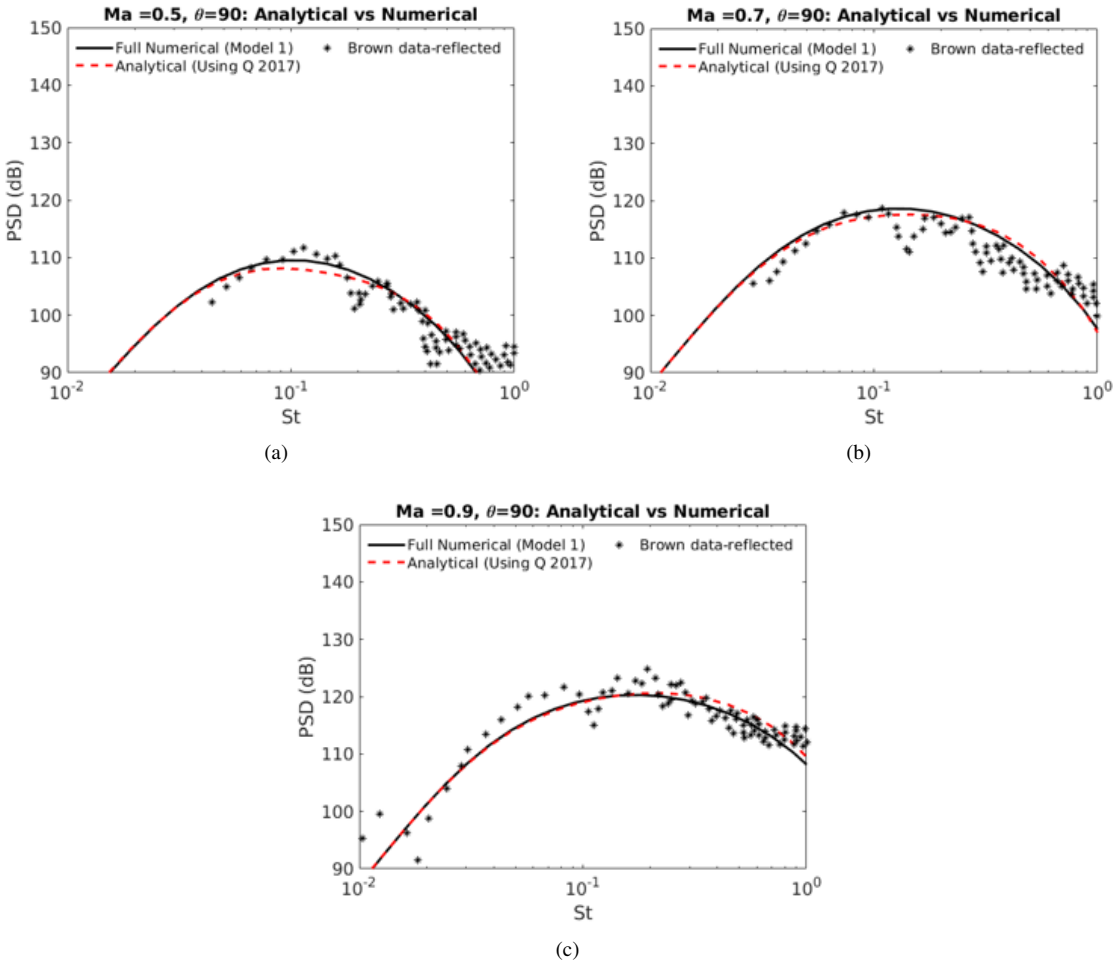
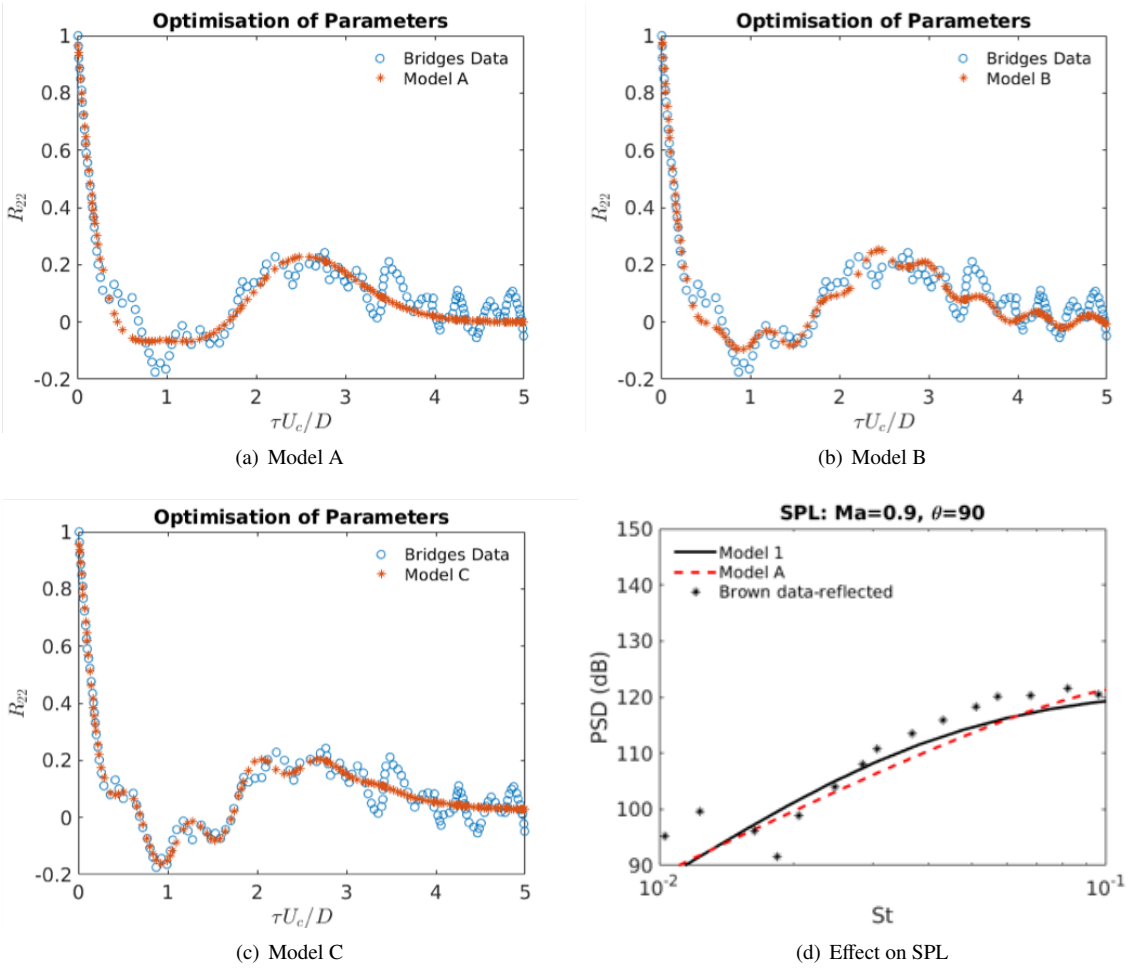


Fig. 14 SPL prediction using analytical formula ($l_3 \to \infty$) against acoustic data in [7]. SPL computed via $SPL = 10log_{10}4\pi(\rho U_j^2)^2 I(\boldsymbol{x};\omega)/P_{ref}^2$ where $P_{ref} = 2\times10^{-5}$ Pa. The spectrum $I(\boldsymbol{x};\omega)$ is determined by integrating (6) over $(u, \tilde{u})$ using (11) and (10).

acoustic predictions given in Fig. (15)d show that for model A, at least, the low frequency roll-off is identical to the full numerical evaluation of (6) (and therefore to the long l_3 limit of (10) when inserted into the Fourier transform (4.22) in GLA19). This indicates at least two pertinent aspects. One, that capturing more complex behaviour of R_{22} does not appear to impact the low frequency decay of the I in (6) and that these features have a greater impact at high frequencies. Second, the spatial decay function that was chosen as a necessary first step to be a simple exponential decay could be reformulated to better capture higher frequencies.

IV. Conclusions

The Goldstein-Leib-Afsar (2019) jet surface interaction model was derived using a self-consistent application of Rapid-distortion theory. The final formula they obtained for the acoustic spectrum of the scattered pressure fluctuation involves the computation of a 4-dimensional integral of the type given by (6). The inner two integrals in the evaluation of the spectrum of the source function (i.e. the convected quantity, $\tilde{\omega}_c$ that enters the basic solution for the pressures fluctuation) require the evaluation of a modified Bessel function of the 2nd kind. Since the latter is usually called via a Library in Fortran 90 computer programming language, the computation of the acoustic spectrum can be prohibitively slow. We have investigated various algorithms (Figs. 10 & 11) to evaluate (6) and (7) by off-loading the inner integrals in (7) to the Graphics-Processing-Unit (GPU) of a computer station. Our calculations show that, in a generic sense,



(a) Model A (b) Model B

(c) Model C (d) Effect on SPL

Fig. 15 R_{22} **prediction against experimental data in [9]. Optimised coefficients are:**
(a) $\mathbf{a_1 = 12.6340, a_2 = 6.2913, a_3 = 5.2367, b = -1.5782, c = 2.0305}$
(b) $\mathbf{a_1 = 12.5603, a_2 = 6.2155, a_3 = 5.2429, a_4 = 0.0355, b = -1.5679, b_2 = -10.4573, c = 2.0530, c_2 = 50.0}$
(c) $\mathbf{a_1 = 1.7854, a_2 = 0.8735, a_3 = -0.1984, a_4 = -0.2196, a_5 = 0.2030 b = -8.5513, b_2 = -0.4033, b_3 = 6.7250,}$
$\mathbf{c = 3.1649, c_2 = 18.2664, c_3 = 1.4039}$
(d) SPL for Model A

when this process (see Fig. 10) is performed to compute a double integral of the type (7), the speed up in the calculation is a factor of 50-times faster than computing it over 1-computer core (Fig. 9).

However, it does not appear to be necessary in the numerical evaluation of the outer integrals in (6) using the actual form of the integrand given by (4.22), (6.26) & (6.27) in GLA19, nor for our model 1 which includes an extra term in R_{22}. These cases converge to 2dp using few iterations and the underlying code is not complex enough to fully utilise the GPU. We find that the benefit of using a GPU is only fully realised when we consider models A-C for R_{22} when we achieve a speed up of up to 30 times, which can increase to 130 times if greater outer iterations are required.

Acknowledgments

MZA would like to thank Strathclyde University for financial support from the Chancellor's Fellowship. SAS would like to thank the EPSRC PhD studentship and the University of Strathclyde/ Mr Colin McLatchie for granting her the Rebecca Fraser Scholarship.

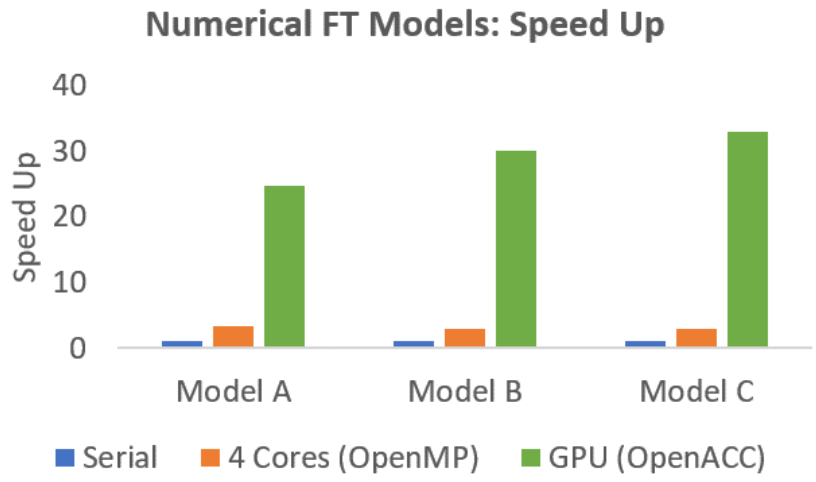


Fig. 16 Speed Up for models including a numerical Fourier transform

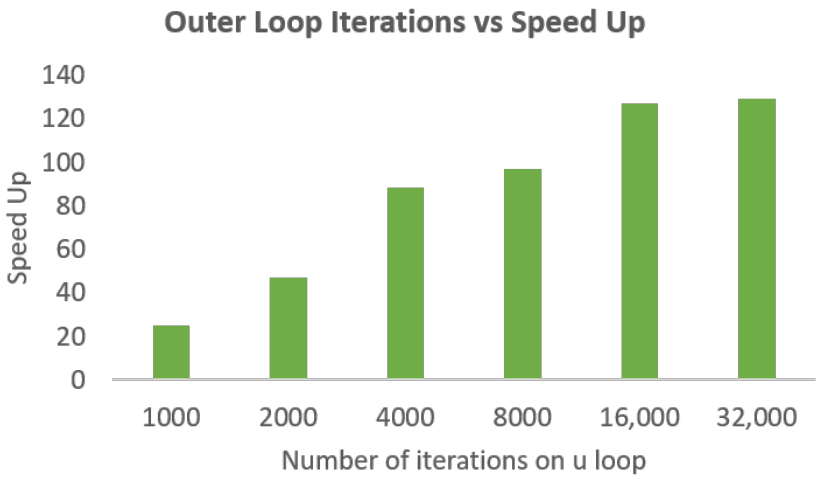


Fig. 17 Model A: Increasing the number of iterations in the outer loop

References

[1] Goldstein, M. E., Leib, S. J., and Afsar, M. Z., "Rapid distortion theory on transversely sheared mean flows of arbitrary cross-section," *J. Fluid Mech.*, Vol. 881, 2019, pp. 551–584.

[2] Bridges, J. E., "Noise from Aft Deck Exhaust Nozzles – Differences in Experimental Embodiments," 21st AIAA/CEAS Aeroacoustics Conference, 2014.

[3] Goldstein, M. E., Afsar, M. Z., and Leib, S. J., "Non-homogeneous Rapid-distortion theory on transversely sheared flows," *J. Fluid Mech.*, Vol. 736, 2013, pp. 532–569.

[4] Goldstein, M. E., Leib, S. J., and Afsar, M. Z., "Generalized rapid distortion theory on transversely sheared mean flows with physically realizable upstream boundary conditions: application to the trailing edge problem," *J. Fluid Mech.*, Vol. 824, 2017, pp. 477–512.

[5] Afsar, M. Z., Leib, S. J., and Bozak, R. E., "Effect of de-correlating turbulence on the low frequency decay of jet-surface interaction noise in sub-sonic unheated air jets using a CFD-based approach," *J. Sound and Vib.*, Vol. 386, 2017, pp. 177–207.

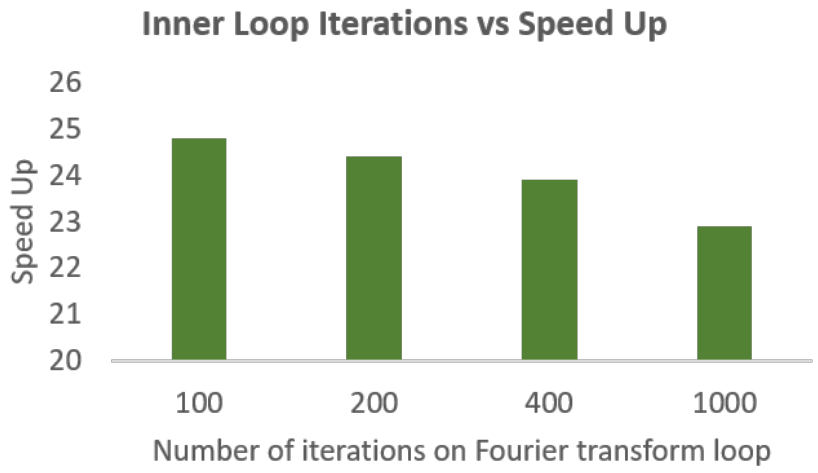


Fig. 18 Model A: Increasing the number of iterations in the inner loop

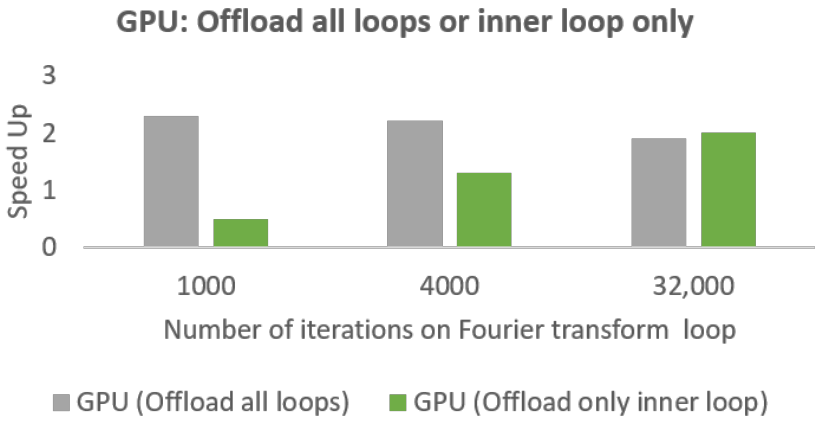


Fig. 19 Model A: Increasing the inner loop iterations to be much greater than the outer loop (outer loop = 100)

[6] Afsar, M. Z., Stirrat, S. A., and Kokkinakis, I. W., "Analytical approximations of the acoustic spectrum for an installed axisymmetric jet flow," 2020 (In prep. for *J. Fluid Mech.*).

[7] Brown, C. A., "Jet-surface interaction test: far-field noise results." *Trans ASME J. Engng Gas Turbines Power*, Vol. 135, 2013.

[8] Afsar, M. Z., Sescu, A., and Minisci, E., "Modeling supersonic heated jet noise at fixed jet Mach number using an asymptotic approach for the acoustic analogy Green's function and an optimized turbulence model," 25th AIAA/CEAS Aeroacoustics Conference, 2019.

[9] Bridges, J. E., "Effect of Heat on Space-Time Correlations in Jets," 12th AIAA/CEAS Aeroacoustics Conference, 2006.

[10] Campbell, G. A., and Foster, R. M., "Fourier integrals for practical applications," Van nostrand, 1954, p. 122.